1. **Summarize for us the goal of this project and how machine learning is useful in trying to accomplish it. As part of your answer, give some background on the dataset and how it can be used to answer the project question. Were there any outliers in the data when you got it, and how did you handle those? [relevant rubric items: "data exploration", "outlier investigation"**

   This dataset describes enron scandal from two aspect: financial features and email features. There are 146 people and each of these people contains 19 features. Among these people, 18 people are person of interest. For features we don't have, we replace the na with 0. Our goal is to predict which one is the person of interest using the information we get. We try to find or create some features which have high correlation with our prediction. It's clearly that some financial data can reflect if a person is guilty since most time the guilty one has more power and money. Meanwhile, it's common that guilty person tends to have chance of connecting other guilty one. That is why our dataset is potential powerful in this problem.

bonus	64	deferral_payments	107
deferred_income	97	director_fees	129
exercised_stock_options	44	expenses	51
from_messages	60	from_poi_to_this_person	60
from_this_person_to_poi	60	loan_advances	142
long_term_incentive	80	other	53
restricted_stock	36	restricted_stock_deferred	128
salary	51	shared_receipt_with_poi	60
to_messages	60	total_payments	21
total_stock_value	20		

Table 1. Numbers of NA

This table shows the situation that the number of NA of different feature. Since we only have 146 samples, it's clear that a lot of features only keep very limited information. We can gain important information from 'loan_advances' since over 95% people don't have valid value. On the other hand, we can find there are only 18 people who are poi (12% of total dataset) which means this is an imbalanced dataset. We need to consider how to split our dataset to get satisfied result.

After visualizing our data, we can find outliers for each feature. First of all, we should drop 'TOTAL' and 'THE TRAVEL AGENCY IN THE PARK' since they are obviously not people. Then we should drop 'LOCKHART EUGENE E' it doesn't contain any feature.

2. **What features did you end up using in your POI identifier, and what selection process did you use to pick them? Did you have to do any scaling? Why or why not? As part of the assignment, you should attempt to engineer your own feature that does not come ready-made in the dataset -- explain what feature you tried to make, and the rationale behind it. (You do not necessarily have to use it in the final analysis, only engineer and test it.) In your feature selection step, if you used an algorithm like a decision tree, please alfso give the feature importances of the features that you use, and if you used an automated feature selection function like SelectKBest, please report the feature scores and reasons for your choice of parameter values. [relevant rubric items: "create new features", "properly scale features", "intelligently select feature"]**

   At beginning of this step, I plotted a table which showed the correlation among all features. According to result, most of features have weak correlation with 'poi' and we need to create new features. I transform some percentage into new features:

   'salary_percentage' = 'salary'/'total_payments'

   'exercised_stock_percentage' = 'exercised_stock_options' / 'total_stock_value'

   'total_value' = 'total_payments' + 'total_stock_value'

   'bonus_of_salary' = 'bonus' / 'salary'

   'future_income' = 'deferral_payments' + 'deferred_income' + 'long_term_incentive'

   'future_income_percentage' = 'future_income' / 'total_payments'

   'from_poi_percentage' = 'from_poi_to_this_person' / 'from_messages'

   'to_poi_percentage' = 'from_this_person_to_poi' / 'to_messages'

   'poi_email'='from_poi_to_this_person'+'from_this_person_to_poi'

   'bonus_percentage' ='bonus'/'total_payments'

   By now, we have 29 usable features and we will drop some overlapped features. For example, the information in ''from_poi_to_this_person' and 'from_messages' can be represented by 'from_poi_percentage'. After dropping some features, we have 24 features in total. Then we apply selectKbest algorithm to see the score for each feature.

exercised_stock_options	24.81	expenses	6.09
total_stock_value	24.18	from_poi_percentage	5.12
bonus	20.79	poi_email	4.86
bonus_percentage	20.71	other	4.18
salary	18.28	to_poi_percentage	4.09
deferred_income	11.45	salary_percentage	2.68
bonus_of_salary	10.78	director_fees	2.12
long_term_incentive	9.922	deferral_payments	0.22
restricted_stock	9.21	future_income_percentage	0.17
total_payments	8.77	future_income	0.072
shared_receipt_with_poi	8.58	restricted_stock_deferred	0.065
loan_advances	7.18	exercised_stock_percentage	0.042

Table 2. Score of different feature

According to this table, we decide to last 3 features and we also find bonus is overlap with bonus_percentage, so 20 features are left in my dataset. Meanwhile, our classifier include PCA and decision tree at last. Since PCA require scaling on the data. So we will scale our features using minmax scaler.

3. **What algorithm did you end up using? What other one(s) did you try? How did model performance differ between algorithms? [relevant rubric item: "pick an algorithm"]**

   I decided to use decision tree as classifier and I also tried Naïve bayes, support vector machine, random forest and AdaBoost Classifier. When I try to use more parameters, all of them meet problem of overfitting except Naïve bayes. Naïve bayes is comparable stable than other algorithm in this case, but we want a better performance by tuning parameters which is why we didn't use it. Decision tree and SVC have higher speed than other two algorithms. We get highest score in train dataset using random forest and AdaBoost Classifier. Accuracy, recall, precision and f1 reached 1. But overfitting is disaster, we only get 0.2 in test dataset using these two algorithm. In our problem, decision tree is a comparable reasonable algorithms.

4. **What does it mean to tune the parameters of an algorithm, and what can happen if you don't do this well? How did you tune the parameters of your particular algorithm? (Some algorithms do not have parameters that you need to tune -- if this is the case for the one you picked, identify and briefly explain how you would have done it for the model that was not your final choice or a different model that**

does utilize parameter tuning, e.g. a decision tree classifier). [relevant rubric item: "tune the algorithm"]

When we apply an algorithm to solve a problem, we need several function to predict result and calculation penalty. When we change the coefficient of these function, we can get different result. Tuning parameters is the process we change it. If we don't try to tune parameters, the classifier will use the default value which may not suitable for your problem.

Different dataset require different parameters to reach higher score. Parameters can control the complexity of classifier. If we apply a complex classifier to a linear-liked simple data, then overfitting can be a big problem. If we use a simple classifier to complex data, we even can't get a good score in training set. By tuning parameter, we can control this trade off.

I used pipeline and gridsearch to tune parameters of my classifier. The pipeline contains PCA and decision tree. For PCA, I tune n_components which means how many component features I want to keep . For decision tree, I tried different criterion: 'gini','entropy'. I also want to check min_samples_split.

5. **What is validation, and what's a classic mistake you can make if you do it wrong? How did you validate your analysis? [relevant rubric item: "validation strategy"]**

   Validation is an important process to see if your algorithm also has good performance for other dataset except train dataset. Overfitting can be a big problem when we don't do validation correctly.

   I used StratifiedShuffleSplit in my code and I set iteration time be 20 to get an average score of my classifier. Test size was set as 0.2.

6. **Give at least 2 evaluation metrics and your average performance for each of them. Explain an interpretation of your metrics that says something human-understandable about your algorithm's performance. [relevant rubric item: "usage of evaluation metrics"]**

   To check the performance of the classifier, I used 2 evaluation metrics: recall, precision. Accuracy is invalid in this problem since our data is imbalanced which mean if we don't do anything, we still can reach a high score. Precision means when the classifier predict 1, the probability that it makes the right choice. Recall means when percentage that how many samples whose true value equals one are predicted by classifier.

   In my classifier, these three metrics can reach over 0.9 for train set and over 0.31 for test set on average. In other words, if this classifier think a person is a person of interest, then the probability that this guy is real guilty person is 31%. This classifier recognizes 31% person of interest from all guilty people.